UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/266,207	03/10/1999	PAUL ENGLAND	777.215US1	5470

22801 7590 11/01/2005

LEE & HAYES PLLC
421 W RIVERSIDE AVENUE SUITE 500
SPOKANE, WA 99201

EXAMINER	
KLIMACH, PAULA W	
ART UNIT	PAPER NUMBER
2135	

DATE MAILED: 11/01/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/266,207	ENGLAND ET AL.	
	Examiner	Art Unit	
	Paula W. Klimach	2135	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 August 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date 8/9.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 08/09/05 has been entered.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-5, 7, 9-13, 15, and 17-18, are rejected under 35 U.S.C. 103(a) as being unpatentable over Angelo (5,944,82 1) in view of Arbaugh and further in view of Anderson (5,974,546).

In reference to claim 1, Angelo discloses a system that comprises a central processing unit (CPU: part 100 Fig. 1 in combination with column 6 lines 8-13) and an operating system (OS), the CPU having a software identity register (Fig. 2 in combination with column 9 lines 35-38), a method for booting the operating system. The secure location is memory and therefore performs the same function as the register of the software identity register. Furthermore Angelo

discloses setting the software identity register to a result of the computed hash value (Fig. 3 and Fig. 4).

Although Angelo discloses saving the hash value (identity of the program) in memory, Angelo does not expressly disclose computing a cryptographic function of at least a portion of the operating system and setting the software identity register to a result of the computed cryptographic function.

Arbaugh discloses a system that verifies the kernel (operating system) by calculating the cryptographic and storing the hash of the operating system level (page 4 section 3.2. 1 paragraph 2 in combination with section 3.2.2 paragraph 4). The cryptographic hash is the identity of the operating system since it is used to verify the integrity of the system.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to calculate the cryptographic hash of the operating system as in Arbaugh in the system of Angelo. One of ordinary skill in the art would have been motivated to do this because calculating the cryptographic hah function is used to calculate the integrity of a function a system is then said to posses integrity, without integrity no system can be made secure (Arbaugh Introduction).

Although Arbaugh discloses a system that verifies the kernel by calculating the cryptographic and storing the hash of the operating system level (page 4 section 3.2. 1 paragraph 2 in combination with section 3.2.2 paragraph 4), the combination of Arbaugh and Angelo do not disclose setting the software identity register to a value indicating that the atomic execution of the boot block failed if the atomic execution of the boot block does not fail.

Anderson discloses a system wherein if the atomic execution of the boot block does not fail, and otherwise setting the software identity register to a value indicating that the atomic execution of the boot block failed (column 5 lines 34-41).

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to set the software identity register of Angelo with a value indicating that the atomic execution of the boot block failed as in Anderson. One of ordinary skill in the art would have been motivated to do this because it would enable the system to determine the cause of a previously failed system boot and based on the analysis, selectively modify specific features and/or system parameters responsive to the cause of the failure during a previous system boot.

In reference to claim 2, Angelo discloses further a method comprising defining a secure storage space, access to which is based in part on the result set in the software identity register (column 9 lines 12-25). The integrity of the huh table is verified by the table hash value stored in the SMM memory.

In reference to claims 3 and 11, Angelo discloses further a system that comprises a central processing unit (CPU: part 100 Fig. 1 in combination with column 6 lines 8-13) and an operating system (OS), the CPU having a software identity register (Fig. 2 in combination with column 9 lines 35-38). The software identity register is a register that stores the identity of related software. A register is a high-speed memory within a microprocessor used to hold data. Angelo discloses setting the software identity register to a result of the computed cryptographic function (Fig. 3 and Fig. 4). Angelo discloses further a system wherein in an event that the operation completes correctly, the software identity register (memory) contains the identity of the program (column 10 lines 16-28) and in an event that the operation fails to complete

correctly, the software identity register contains a value other than the identity of the program; and examining a content of the software identity register to verify the identity of the program(column 10 lines 39-65). The hash value can be deleted; this would be setting the value to something other than the correct hash value. The user is also given a choice to update the value and put in a value that is different from the correct hash value.

However Angelo does not expressly disclose the identity of the software being the identity of the operating system.

Arbaugh discloses a system that verifies the kernel (operating system) by calculating the cryptographic hash of the operating system level (page 4 section 3.2. 1 paragraph 2 in combination with section 3.2.2 paragraph 4). The cryptographic hash is the identity of the operating system since it is used to verify the integrity of the operating system. The system of Arbaugh also expressly discloses a system for loading the operating system (Figure 3).

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to calculate the cryptographic hash of the operating system as in Arbaugh in the system of Angelo. One of ordinary skill in the art would have been motivated to do this because calculating the cryptographic hash function is used to calculate the integrity of a function a system is then said to posses integrity, without integrity no system can be made secure (Arbaugh Introduction).

Although Arbaugh discloses a system that verifies the kernel by calculating the cryptographic and storing the hash of the operating system level (page 4 section 3.2. 1 paragraph 2 in combination with section 3.2.2 paragraph 4), the combination of Arbaugh and Angelo do not

disclose setting the software identity register to a value indicating that the atomic execution of the boot block failed if the atomic execution of the boot block does not fail.

Anderson discloses a system wherein if the atomic execution of the boot block does not fail, and otherwise setting the software identity register to a value indicating that the atomic execution of the boot block failed (column 5 lines 34-41).

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to set the software identity register of Angelo with a value indicating that the atomic execution of the boot block failed as in Anderson. One of ordinary skill in the art would have been motivated to do this because it would enable the system to determine the cause of a previously failed system boot and based on the analysis, selectively modify specific features and/or system parameters responsive to the cause of the failure during a previous system boot.

In reference to claims 4, 9, 10, 12, 17, and 18, the identity comprises a public key of a correctly signed block of code from the operating system, and examining a content of the software identity register comprises verifying a signature of the signed block of code against the public key (Section 3.2.2 paragraph 2 Arbaugh).

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to calculate the cryptographic hash of the operating system as in Arbaugh in the system of Angelo. One of ordinary skill in the art would have been motivated to do this because calculating the cryptographic hash function is used to calculate the integrity of a function a system is then said to posses integrity, without integrity no system can be made secure (Arbaugh Introduction).

In reference to claims 7 and 15, that further comprises the authentication of additional blocks of code.

Arbaugh teaches authenticating sections of code using a signature (page 4 Section 3.2.2 paragraph 2).

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to calculate the cryptographic hash of the operating system as in Arbaugh in the system of Angelo. One of ordinary skill in the art would have been motivated to do this because calculating the cryptographic hash function is used to calculate the integrity of a function a system is then said to posses integrity, without integrity no system can be made secure (Arbaugh Introduction).

In reference to claims 5 and 13 are rejected as in rejection for claims 3 and 11. Angelo discusses a hash value generated by an integrity assessment code that is specific to a given software application although the disclosed embodiment of the invention utilizes a hash table 206 containing hash values generated by a secure hash algorithm 208; it is contemplated that many types of modification detection codes could be utilized. Of importance to the invention is that each piece of software to be tracked has a corresponding and fairly unique value that represents the unaltered state of the software, and that this value be stored in a secure memory location (Fig. 3).

Claim 6, 8, 14, 16, and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Angelo, Arbaugh, and Anderson as applied to claims 3, 11, 19 are respectively above, and further in view of Sadowsky et al (6,230,285 B1).

In reference to claims 6, 8, 14, and 16, Angelo does not expressly disclose maintaining a boot log.

Sadowsky discloses maintaining a boot log (Fig 4). Further Sadowsky suggest the boot file comprising appending at least a portion of the identity to a boot log (column 4 lines 65 and 66).

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to append the identity to the boot log of Sadowsky in the system of Angelo. One of ordinary skill in the art would have been motivated to do this because it will show the cause of boot failure (column 5 lines 12-15).

In reference to claims 21, the method wherein creating an identity of the OS comprises forming the OS certificate with one or more items from a boot log containing identities of software components that are executing on the CPU. The boot log discussed by Sadowsky contains information such as the device driver and executables (column 4 lines 65 and 66). This information is shared with the certificate information suggested by Barr.

Claim 19 is rejected under 35 U.S.C. 103(a) as being unpatentable over Angelo in view of Arbaugh and further in view of Stallings (Cryptography and Network Security).

In reference to claim 19, Angelo teaches a system that includes a CPU (part 100 Fig. 1 in combination with column 6 lines 8-13) and an operating system (OS), the CPU having a software identity register (Fig. 2 in combination with column 9 lines 35-38). In addition, Angelo discloses a system wherein in an event that the operation completes correctly, the software identity register contains the identity of the operating system (column 10 lines 16-28) and in an event that the

operation fails to complete correctly, the software identity register contains a value other than the identity of the operating system; and examining a content of the software identity register to verify the identity of the operating system (column 10 lines 39-65). The hash value can be deleted; this would be setting the value to something other than the correct hash value. The user is also given a choice to update the value and put in a value that is different from the correct hash value.

However, Angelo does not expressly disclose having a pair of private and public keys and a software identity register that holds an identity of the operating system. The identity of the software created in Angelo is not expressly disclosed as the identity of the OS containing the and signing the OS certificate using the CPU private key.

Arbaugh discloses a system that verifies the kernel (operating system) by calculating the cryptographic hash of the operating system level (page 4 section 3.2. 1 paragraph 2 in combination with section 3.2.2 paragraph 4). The cryptographic hash is the identity of the operating system since it is used to verify the integrity of the operating system. Arbaugh also teaches the use of digital signatures and public key certification, therefore the use of private and public keys (page 4 section 3.2. 1 paragraph 1).

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to calculate the cryptographic hash of the operating system as in Arbaugh in the system of Angelo. One of ordinary skill in the art would have been motivated to do this because calculating the cryptographic hash function is used to calculate the integrity of a function a system is then said to posses integrity, without integrity no system can be made secure (Arbaugh Introduction).

Although Arbaugh discloses a system that verifies the kernel by calculating the cryptographic hash of the operating system level (page 4 section 3.2. 1 paragraph 2 in combination with section 3.2.2 paragraph 4), Arbaugh does not expressly disclose a verification system that users certificates.

Stallings discloses verification using digital certificates (pages 186-187). The OS certificate would be signed using the CPU private key if the private key of the CPU is the same as the private key of the CPU.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to utilize digital certificates for the verification process of Stalling instead of the system disclosed by Arbaugh. One of ordinary skill in the art would have been motivated to do this because any participant can verify that the certificate originated from the certificate authority and is not counterfeit.

Claims 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over Angelo, Arbaugh, and Stallings as applied to claims 19 above, and further in view of LeBourgeois (6,026,166).

LeBourgeois further suggests submitting the signed software identity register (the identity of the user) over a network to a third party to prove an identity of the operating system to the third party (Fig 3A and Fig. 3B).

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to bind the identification of the device drive to the signature of the certificate as in LeBourgeois in the system of Angelo. One of ordinary skill in the art would have been motivated

to do this because it is useful in ensuring that digital products are authorized for use on only one machine (column 3 lines 21-23).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Paula W. Klimach whose telephone number is (571) 272-38544. The examiner can normally be reached on Mon to Thr 9:30 a.m to 5:30 p.m.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kim Vu can be reached on (571) 272-3859. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

PWK
Wednesday, October 26, 2005

Primary Examiner
Art Unit 2135